UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/029,496	12/21/2001	Sivaram Krishnan		4639

39170 7590 12/16/2005

HITACHI AMERICA, LTD.
50 PROSPECT AVENUE
LEGAL DEPT.
TARRYTOWN, NY 10591

EXAMINER	
ROMANO, JOHN J	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 12/16/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/029,496	KRISHNAN, SIVARAM	
	Examiner	Art Unit	
	John J. Romano	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 September 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 and 3-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 and 3-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Remarks

1. Applicant's amendment and response received September 22nd, 2005, responding to the May 19th, 2005, Office action provided in the rejections of claims 1 and 3-24, wherein claims 1, 4, 15, 17 and 22 have been amended. Claims 1 and 3-24 remain pending in the application and which have been fully considered by the examiner.

Applicant arguing for the claims being patentable over *Subramanian* and *Roediger* (see pages 7-9 of the amendment and response), and arguments pertaining to the dependent claims are not persuasive, as will be addressed under Prior Art's Arguments – Rejections section at item 2 and the claim rejections below. Accordingly, Applicants' amendment necessitated additional clarifications, in light of the rejection of the claims over prior art provided in the previous Office action, to further point out that the prior art also discloses as such claimed limitations as now amended which will be provided and/or addressed under the item 2 below. Thus, the rejection of the claims over prior art in the previous Office action is maintained in light of the necessitated additional clarifications provided hereon and **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Prior Art's Arguments – Rejections

2. Applicant's arguments filed September 22nd, 2005, in particular on pages 7-8, have been fully considered but they are not persuasive. For example,

(A) In regard to the argument that *Subramanian* does not disclose adding an internal access point to handle statements unnecessarily iterated, e.g., "loop-invariant" or "omega-invariant" statements, (page 8, second paragraph of the amendment and response), the examiner respectfully disagrees. Applicant bases the above conclusion on the assertion that the kernel (503), is not an access point (Page 7, Fourth Paragraph); then concludes that the Kernel is the iterative loop itself without an internal access point. The examiner would like to respectfully draw Applicant's attention to Column 6, lines 46-58, wherein *Subramanian* discloses that the compiled instructions that implement loops are organized into a prologue and a kernel as reproduced below:

- "The *compiled instructions* that implement loops are organized into a prologue that initializes the loop and a kernel that contains the opcodes representing the compiled statements within the loop's body and the test for the completion condition on the

loop. " (E.g., see Column 6, lines 44-48), wherein the compiler implements a prologue.

By adding the prologue the access point of the main loop becomes an internal access point as the prologue is part of the loop as disclosed by *Subramainian* (Column 6, line 62 – Column 7, line 8), wherein the optimized version of the loop consists of the peels, omega-invariant operation and the kernel. In the optimized version, the loop code is segmented into segments and the original access point of the kernel is an internal access point relative to the omega-variant peels that are before it, and also part of the original loop. Therefore, the access point of the kernel is indeed an internal access point as disclosed by *Subramanian*. Thus, the rejection is maintained in respect to the instant argument.

(B) In regard to the argument that *Roediger* does not disclose automatically adding any instructions to the loop, let alone an "internal access point" to handle an unnecessarily iterated statement and let alone more specifically an "internal recursive entry point" (page 8, third paragraph), the Examiner would like to point out that *Roediger* was only used for adding an internal recursive entry point. As previously indicated, *Roediger* discloses (E.g., see Page 3, Figure 4 & Paragraph [0043]), an internal access point, which is called recursively as shown in 420, wherein the access point is internal when code from the original loop is peeled and placed outside the original loop to create a sequence of segmented code, wherein code is repeated. For a further example of the method, see Paragraph [0045], wherein *Roediger* discloses the peeled iterations which are hoisted out of the main body of the loop, thereby creating an internal access

point that did not exist before the optimization. Therefore, *Roediger* teaches an "internal recursive entry point".

In regard to Applicant's argument with respect to "...automatically...", the argument is moot in view of new grounds of rejection in view of *Subramanian*, wherein *Subramanian's* method is programmatic and thus automatic. Therefore, *Roediger* is not used to teach "...automatically..." as disclosed herein, and below in the claim rejections.

Claim Rejections

3. Claims 1 and 3-24, are pending claims, stand finally rejected in light of the additional clarifications provided and/or addressed at item 2 above, Prior Art's Arguments – Rejections.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless-
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims **1, 5-7, 10, 11, 14, 15** and **17-24** are rejected under 35 U.S.C. 102(b) as being clearly anticipated by Subramanian, US 6,026,240, (hereinafter **Subramanian**).

As per Claim **1**, **Subramanian** discloses a method for improving execution performance in a sequence of instructions that provide a function and having external access points that include at least one external entry point and at least one external exit point, comprising the steps of:

"...*determining... at least one instruction, from the sequence of instructions that is necessary to be executed for less than all repetitions of the sequence of instructions; and...*" (E.g., see Figure 5A (501) & Column 11, lines 4-21), wherein the prologue (501) includes at least one instruction from the sequence which are necessary to be executed for less than all repetitions of the sequence of instructions;

"...*automatically...*" (E.g., see Figure 2 & Column 4, lines 48-57), wherein the process is performed programmatically (automatic) by a compiler;

"...*adding automatically an internal access point to the sequence of instructions to isolate the one instruction from only some of the repetitions of the sequence of instructions, thereby partitioning the sequence of instructions into multiple segments, and having one of the multiple segments including the one instruction and executing for fewer times than the number of executions of another of the multiple segments .*" (E.g., see Figure 5A (503) & Column 11, lines 4-21), wherein the kernel (503) is an internal access point added to the sequence of instructions, thereby partitioning the sequence into multiple segments.

3. In regard to claim **5, Subramanian** discloses an internal access point (E.g., See Fig. 5a & Col. 11, lines 4-16), where the "prologue" is the external access point and the

"kernel" is the internal access point. Furthermore, **Subramanian** discloses, *"...moving the one instruction from outside of the one of the multiple segments to within the one of the multiple segments..."* (E.g., See Fig. 5b, 509 & Col. 11, lines 28-34). Where the "loop-invariant operations" are contained in the prologue, which is between the external access point and the internal access point.

4. In regard to claim **6, Subramanian** also discloses "*...said modifying includes the step of rescheduling the one instruction closer in sequence of execution to one of the external access points."* (E.g., See Fig. 5b & Col. 11, lines 29-42), where the hoisted loop-invariant instructions are executed in the prologue. It is noted that the prologue is the entry point from the external access, thereby, re-arranging or rescheduling the one instruction closer in sequence of execution to one of the external access points.

5. In regard to claim **7, Subramanian** discloses "*A computer readable storage media having computer readable code physically implementing a method of improving execution performance of a sequence of instructions..."* (E.g., See Fig. 1 & Col. 8, lines 33-42), wherein the embodiment of the invention is a method of improving execution performance of a sequence of instructions, (E.g., See Col. 4, lines 48-57).

6. In regard to claim **10,** this is a computer readable storage media version of the claimed method discussed above, in claim **6**, wherein all claimed limitations have also been addressed and/or cited as set forth above. **Subramanian** discloses the above limitation (E.g., See Fig. 1 & Col. 8, lines 33-42).

7. In regard to claims **11** and **14,** this is a computer system version of the claimed method discussed above, in claims **7 and 10**, wherein all claimed limitations have also

been addressed and/or cited as set forth above. For example, **Subramanian** discloses, "...at least one processing unit coupled to said computer readable storage media...", (E.g., See Fig. 1 & Col. 8, lines 33-42), where the "CPU" is interpreted as coupled to the stated "CD-ROM medium that typically contains programs". The CD-ROM medium is a non-volatile memory. Furthermore, **Subramanian** discloses "*...volatile memory*", (E.g., See Col. 14, lines 42-46).

8. In regard to Claim **15, Subramanian** discloses, "*A method of machine executing a called program of a repeated sequence of instructions having at least one instruction that is necessary to be executed for less than all repetitions of the program comprising:*

"...executing at least some of the sequence of instructions from an externally called entry point; thereafter repeatedly calling the sequence; in response to repeatedly calling, executing only some of the sequence of instructions..." (E.g., see Figure 5A (501) & Column 11, lines 4-21), wherein the prologue (501) includes at least one instruction from the sequence which are necessary to be executed for less than all repetitions of the sequence of instructions and is executed from an externally called entry point (prologue) thereafter executing only some of the instructions (kernel);

"...thereafter executing the sequence from an exit point; and *controlling...at least one of said steps of executing with an internal access point other than the entry point and the exit point to isolate at least the one instruction within the sequence of instructions from as least one of said repeatedly calling and to execute the one instruction a number of times fewer than the total number of executions of the sequence of instructions.*" (E.g., see Figure 5A (501) & Column 11, lines 4-21), wherein the

kernel, wherein the kernel comprises a header or label, which is the internal access point, and executes the remaining sequence of instructions and exits via an external exit point.

"...*automatically*..." (E.g., see Figure 2 & Column 4, lines 48-57), wherein the process is performed programmatically (automatic) by a compiler;

9. In regard to Claim **17, Subramanian** discloses, "*A method processing comprising:*

"...providing a sequence of instructions repeatable to perform a function and having at least one instruction that is necessary to be executed for less than all repetitions of the sequence of instructions..." (E.g., see Figure 5A (501) & Column 11, lines 4-21), wherein the prologue (501) includes at least one instruction from the sequence which are necessary to be executed for less than all repetitions of the sequence of instructions and is executed from an externally called entry point (prologue) thereafter executing only some of the instructions (kernel);

"...*automatically*..." (E.g., see Figure 2 & Column 4, lines 48-57), wherein the process is performed programmatically (automatic) by a compiler;

"...providing *an internal access point other than the entry point and an external exit point, which internal access point isolates the one instruction within the sequence of instructions form only some of the repetitions so that the one instruction is within less than all of the repetitions.*" (E.g., see Figure 5A (501) & Column 11, lines 4-21), wherein the kernel, comprises a header or label, which is the internal access point, and executes

the remaining sequence of instructions, thereby isolating the one instruction from only some of the repetitions and exits via an external exit point.

10. In regard to claims **18, 19 and 20, Subramanian** disclose the method of claim **17** as disclosed above. Further, **Subramanian** discloses "...at least one processing unit coupled to said computer readable storage media...", (E.g., See Fig. 1 & Col. 8, lines 33-42), where the "CPU" is interpreted as coupled to the "CD-ROM medium that typically contains programs", since a processor is required to make the program effectively operable. The CD-ROM medium is a non-volatile memory. Furthermore, **Subramanian** discloses "*...volatile memory*", (E.g., See Col. 14, lines 42-46). In either case, the act of the CPU executing an instruction to store the program from volatile to non-volatile memory is well known by those in the programming art. Thus, meeting the further limitation of "*...wherein all of said steps are included within a step of storing a program" or "...within transmitting a program", or "...receiving the program.",* the data of the program instructions being transmitted via the data-bus via direction from the CPU and furthermore, the act of the memory receiving the program from the data-bus.

11. In regard to claims **21 and 22**, **Subramanian** discloses the method of claim **17** as disclosed above in claim **17**. Further, **Subramanian** additionally discloses the above limitations, (E.g., See Col. 14, lines 42-46). In either case, the act of the CPU executing an instruction to compile the program from volatile or non-volatile memory is well known by those in the programming art as modifying, (via compiling), and executing the program, (performing the steps of the process via CPU). Thus, meeting the further

limitation of "...*wherein all of said steps are included within a step of executing a program", or "...within a step of machine modifying a program."*

12. In regard to claim **23**, **Subramanian** discloses the method of claim **6** as disclosed above. Furthermore, **Subramanian** discloses "...comprising a storage media;...", (E.g., See Fig. 1 & Col. 8, lines 33-42), as stated above, where the CD-ROM medium is a storage media.

13. In regard to claim **24**, **Subramanian** disclose the method of claim **23** as disclosed above. **Subramanian** further discloses "...said means for rescheduling providing internal recursive access between an entry point and an exit point of the sequence of instructions...", (E.g., See Fig. 5b & Col. 11, lines 29-42), where the hoisted loop-invariant instructions are executed in the prologue. It is noted that the prologue is the entry point from the external access, thereby, re-arranging or rescheduling the one instruction closer in sequence of execution to one of the external access points.

Claim Rejections - 35 USC § 103

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15. Claims **3, 4, 8, 9, 12, 13** and **16** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Subramanian** in view of Roediger et al., US 2003/0097652 (hereinafter **Roediger**).

16. As per claim **3**, **Subramanian** teaches the method of claim **1** as described above. But **Subramanian** does not expressly disclose *"...wherein said adding inserts the one internal access point as an internal recursive entry point."* However, **Roediger** discloses:

- *"...wherein said adding inserts the one internal access point as an internal recursive entry point."* (E.g., see Figure 4, Page 3, Paragraph [0043], wherein an internal access point is added as an internal recursive entry point.

Roediger and **Subramanian** are analogous art because they are both concerned with the same field of endeavor of optimizing programs including loops. Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify **Subramanian's** compiler with **Roediger's** internal recursive entry point. The motivation would be to optimize execution speed (**Subramanian**, Column 2, line 16) and thus improve efficiency.

17. In regard to claim **4, Subramanian** also discloses:

"...and the internal recursive entry point." Where the "loop-invariant operations" are contained in the prologue, (E.g., See Fig. 5b & Col. 11, lines 9-13), which is between the external access point and the internal access point. Furthermore each

"kernel" can be an iterative process, (E.g., See Fig. 5b & Col. 11, lines 13-21), which calls itself and therefore is an internal recursive access point, wherein the loop is entered by the access point.

18. In regard to claim **8, Subramanian** also discloses:

"A computer readable storage media having computer readable code physically implementing a method of improving execu*tion performance of a sequence of instructions..."* (E.g., See Fig. 1 & Col. 8, lines 33-42), as stated above, where the embodiment of the invention is a method of improving execution performance of a sequence of instructions, (E.g., See Col. 4, lines 48-57).

19. In regard to claim **9**, **Subramanian** discloses the method of claim **5** as disclosed above. But, in claim **5**, **Subramanian** does not expressly disclose "...of a recursive sequence of instructions.". However, **Roediger** discloses "...of a recursive sequence of instructions.", as discussed in claim **3** above.

20. In regard to claims **12** and **13** this is a computer system version of the claimed method discussed above, in claims **8** and **9**, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, **Subramanian** discloses, "...at least one processing unit coupled to said computer readable storage media...", (E.g., See Fig. 1 & Col. 8, lines 33-42), where the "CPU" is interpreted as coupled to the stated "CD-ROM medium that typically contains programs". The CD-ROM medium is a non-volatile memory. Furthermore, **Subramanian** discloses "*...volatile memory*", (E.g., See Col. 14, lines 42-46).

21. In regard to claim **16**, the method of base claim **15** is incorporated. Furthermore, **Subramanian** discloses, *"...said first-mentioned executing, includes executing the one instruction..."* (E.g., See Fig. 5b & Col. 11, lines 29-42), as stated above, where the hoisted loop-invariant instructions are executed upon first execution, which is the prologue execution. **Subramanian** also discloses, "*...said second-mentioned executing recursively starts form the internal recursive entry point.* (E.g., See Fig. 5a & Col. 11, lines 4-16), where the kernel is the internal recursive entry point upon which second execution starts. But **Subramanian** does not expressly disclose *"...said internal access point is an internal recursive entry point scheduled after the one instruction in the sequence of instructions..."* However, **Roediger** discloses:

- *"...said internal access point is an internal recursive entry point scheduled after the one instruction in the sequence of instructions..."* (E.g., see Figure 4, Page 3, Paragraph [0043], wherein an internal access point is added as an internal recursive entry point scheduled after the one instruction.

Roediger and **Subramanian** are analogous art because they are both concerned with the same field of endeavor of optimizing programs including loops. Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify **Subramanian's** compiler with **Roediger's** internal recursive entry point. The motivation would be to optimize execution speed (**Subramanian**, Column 2, line 16) and thus improve efficiency.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to John J. Romano whose telephone number is (571) 272-3872. The examiner can normally be reached on 8-5:30, M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JJR

TUAN DAM
SUPERVISORY PATENT EXAMINER